**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

***Escuela Profesional Académica de Ingeniería de Software***

**SISTEMA DE PLANIFICACIÓN DE ESTUDIOS UNIVERSITARIOS - FOLLOW CLASS (FC)**

**PLAN DE PROYECTO**

**Documento de Arquitectura de Software**

**GRUPO 6**

**DOCENTE:** Dra. Lenis Rossi Wong Portillo

**CURSO:** Gestión de la Configuración del Software

**COORDINADORA:** Romero Diaz, Bianca Elizabeth

**INTEGRANTES:**

Balandra Camacho, Ivan 20200248

Hernández Bianchi, Stefano Alessandro 20200309

Marcelo Salinas, Moises Enrique 20200310

Ortiz Crisostomo, Edwin Jose 14200224

Quispe Fajardo, Adrián Ismael 20200281

Romero Diaz, Bianca Elizabeth 20200312

Solis Flores, Aldair Jhostin 20200293

**2022-I**

**HISTORIAL DE CAMBIOS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Autor(es)** | **Descripción** | **Fecha** |
| ***1.0*** | **EQUIPO 6**  Hernández Bianchi, Stefano  Romero Diaz, Bianca Elizabeth  Marcelo Salinas, Moises  Quispe Fajardo, Adrián  Solis Flores, Aldair | *-Documentación de Arquitectura de Software*  *-Diseño de Software* | 09.06.22 |
| ***2.0*** | **EQUIPO 6**  Hernández Bianchi, Stefano  Romero Diaz, Bianca Elizabeth  Marcelo Salinas, Moises  Quispe Fajardo, Adrián  Solis Flores, Aldair | *-Actualización de Impacto de Software por Módulos* | 14.07.22 |

**Índice**

Contenido

[1. Introducción 4](#_Toc108742899)

[1.1. Propósito 4](#_Toc108742900)

[1.2. Alcance 4](#_Toc108742901)

[1.3. Objetivos 4](#_Toc108742902)

[2. Consideraciones generales 4](#_Toc108742903)

[2.1. Definición de la Arquitectura 4](#_Toc108742904)

[2.2. Tecnologías utilizadas 5](#_Toc108742905)

[2.3. Capas arquitectónicas 5](#_Toc108742906)

[2.3.1. Capa de Módulo: 6](#_Toc108742907)

[2.3.3. Capa controlador: 6](#_Toc108742908)

[DISEÑO DE SOFTWARE 6](#_Toc108742909)

[1. INTRODUCCIÓN 6](#_Toc108742910)

[1.1. Propósito 6](#_Toc108742911)

[1.2. Alcances 7](#_Toc108742912)

[1.3. Objetivo 7](#_Toc108742913)

[2. IMPACTO DEL SOFTWARE POR MÓDULOS 7](#_Toc108742914)

[2.1. Módulo Autentificación de usuario 7](#_Toc108742915)

[2.1.1. Tabla entidad estudiante 7](#_Toc108742916)

[2.1.2. Interfaz de usuario 8](#_Toc108742917)

[2.2. Módulo Usuario Cliente 10](#_Toc108742918)

[2.2.1. Tabla entidad curso 10](#_Toc108742919)

[2.2.2. Tabla entidad tareas 10](#_Toc108742920)

[2.2.3. Interfaz de Usuario 11](#_Toc108742921)

**DOCUMENTACIÓN DE ARQUITECTURA DE SOFTWARE**

# Introducción

## Propósito

El propósito de este documento es mostrar la arquitectura del software que se ha planteado para la aplicación de escritorio de Follow Class. Es así que el documento será de gran utilidad para los analistas y arquitectos de software que desarrollen la aplicación.

## Alcance

* Indicar las tecnologías a emplear para el proyecto Follow Class
* Desarrollar la arquitectura del proyecto Follow Class

## Objetivos

* Establecer una base para los analistas y desarrolladores con la que podrán verificar la arquitectura del software planteada de acuerdo con los requisitos previamente indicados.
* Permitir a los analistas y desarrolladores modificar la arquitectura planteada para que estos logren adaptarse de manera más adecuada a los requisitos.
* Especificar las tecnologías que servirán de soporte para el desarrollo del Software y su utilidad en la estructura arquitectónica del aplicativo.

# 2. Consideraciones generales

## 2.1. Definición de la Arquitectura

Para el desarrollo del programa Follow Class, se trabajó con los niveles de presentación, aplicación y datos. De esta manera se puede trabajar de forma modular y mantenible. Además, separar las funciones de la aplicación en su interfaz, su nivel lógico y el backend hace que la aplicación sea muy ligera.

## 2.2. Tecnologías utilizadas

|  |  |  |
| --- | --- | --- |
| Tecnología | Descripción | Logo |
| Apache NetBeans IDE 12.6 | Entorno de desarrollo integrado libre que proporciona soporte para lenguajes de programación como Java, PHP, C/C++, entre otros. Se utilizará para la codificación y el desarrollo de interfaces gráficas del aplicativo, debido a que proporciona esta funcionalidad. | Forma  Descripción generada automáticamente |
| Java 18.0.1 | Java es un lenguaje de programación orientado a objetos. Se utilizará para el control de la aplicación. Además se usará su biblioteca gráfica Swing la cual incluye widgets para interfaz gráfica de usuario. | Logotipo, nombre de la empresa  Descripción generada automáticamente |
| pgAdmin 4 version 6.9 | Herramienta GUI de gestión y administración de base de datos PostgreSQL. Se utilizará para interactuar con las sesiones de la base de datos, tanto en servidores locales como remotos. | Icono  Descripción generada automáticamente |
| Heroku Postgres | Es un servicio de base de datos  PostgresSQL administrado y proporcionado por Heroku. Se utilizará este servicio debido a su escalabilidad y confiabilidad de gestión de los datos con las aplicaciones. | Icono  Descripción generada automáticamente |

## 2.3. Capas arquitectónicas

A continuación, se muestra el diagrama de arquitectura del software:

**Imagen que contiene Logotipo

Descripción generada automáticamente**

### 2.3.1. Capa de Módulo:

El nivel de datos, a veces denominado nivel de base de datos, nivel de acceso a datos o backend, es donde se almacena y gestiona la información procesada por la aplicación. Puede ser un sistema de gestión de base de datos relacional como PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix o Microsoft SQL Server, o en un servidor de bases de datos NoSQL como Cassandra, CouchDB o MongoDB.

2.3.2. Capa de Vista:  
 El nivel de presentación es la interfaz de usuario y de      comunicación de la aplicación, donde el usuario final interactúa con la aplicación. Su objetivo principal es mostrar información al usuario y recopilar datos de este. Este primer nivel se puede ejecutar en un navegador web como una aplicación de desktop o una interfaz gráfica de usuario (GUI).

### 2.3.3. Capa controlador:

El nivel de aplicación, también conocido como el nivel lógico o medio, es el núcleo de la aplicación. En este nivel, se procesa la información recopilada en el nivel de presentación, a veces con otra información en el nivel de datos, mediante la lógica empresarial; un conjunto específico de reglas empresariales. El nivel de aplicación también puede añadir, suprimir o modificar datos en el nivel de datos.

# DISEÑO DE SOFTWARE

# 1. INTRODUCCIÓN

## 1.1. Propósito

La actual documentación posee los componentes de diseño y una lista de los posibles impactos en el sistema a causa de la previa implementación de sus componentes.

## 1.2. Alcances

* Identificar los impactos dentro del sistema y organizarlos por cada módulo
* Enlistar aquellos procesos que se verán afectados luego de la aplicación del proyecto al sistema.

## 1.3. Objetivo

Simplificar la lista de procedimientos y/o procesos para su posterior análisis del área de sistemas y su correcto desarrollo.

# 2. IMPACTO DEL SOFTWARE POR MÓDULOS

En esta sección daremos a conocer una lista de los procesos y/o componentes del sistema que se verán afectados al momento de realizar la implementación del proyecto.

Esta información puede ser de ayuda para el área de desarrollo para identificar los procesos que se deben implementar dentro del proyecto

Primero, se realiza un resumen del impacto por módulo y producto en la implementación del proyecto (ver la siguiente Tabla: Impacto por módulo y producto del proyecto Follow Class).

|  |  |  |
| --- | --- | --- |
| **PRODUCTO** | **MÓDULO** | **NIVEL DE IMPACTO** |
| ***Follow Class(FC)*** | ***Autenticación de Usuario*** | Alto |
| ***Usuario Cliente*** | Alto |

## 2.1. Módulo Autentificación de usuario

### 2.1.1. Tabla entidad estudiante

|  |  |  |
| --- | --- | --- |
| **ENTIDAD: ESTUDIANTE** | | |
| **CAMPO** | **DESCRIPCIÓN** | **TIPO DE DATO** |
| id\_estudiante | Identificador del estudiante | int |
| nombreApellido | Nombre y apellido del estudiante | varchar (45) |
| email | Correo electrónico | varchar (45) |
| contraseña | Contraseña del estudiante | varchar (45) |
| facultad | Facultad del estudiante | varchar (45) |
| escuela | Escuela del estudiante | varchar (45) |
| plan | Plan del estudiante | varchar (45) |

### 2.1.2. Interfaz de usuario

#### 2.1.2.1. Registrarse

La aplicación Follow Class, contará con una vista para poder registrar un nuevo usuario.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

#### **2.1.2.2. Iniciar sesión**

La aplicación Follow Class contará con una vista para que los usuarios registrados puedan acceder a su cuenta

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## 2.2. Módulo Usuario Cliente

### 2.2.1. Tabla entidad curso

|  |  |  |
| --- | --- | --- |
| **ENTIDAD: CURSO** | | |
| **CAMPO** | **DESCRIPCIÓN** | **TIPO DE DATO** |
| id\_curso | Identificador del curso | int |
| nombre | Nombres del curso | varchar (45) |
| docente | Docente del curso | varchar (45) |
| facultad | Facultad del curso | varchar (45) |
| escuela | Escuela del curso | varchar (45) |
| plan | Plan del curso | varchar (45) |
| periódo\_académico | Periodo académico del curso | varchar (45) |
| fecha\_inicio | Fecha inicio del curso | datetime |
| fecha\_fin | Fecha fin del curso | datetime |
| cantidad\_semanas | Cantidad de semanas del curso | int |
| ciclo | Ciclo perteneciente al curso | int |
| tipo | Tipo de curso | varchar(45) |
| créditos | Créditos del curso | int |
| id\_programación | Identificador de programación | int |

### 2.2.2. Tabla entidad tareas

|  |  |  |
| --- | --- | --- |
| **ENTIDAD: TAREAS** | | |
| **CAMPO** | **DESCRIPCIÓN** | **TIPO DE DATO** |
| id\_tarea | Identificador de programación | int |
| título | Día de programación | varchar (45) |
| fecha | Fecha de la programación | date |
| hora\_inicio | Hora de inicio de la programación | time |
| hora\_fin | Hora de fin de la programación | time |
| color\_r | Valor R en calendario | Int |
| color\_g | Valor G en calendario | Int |
| color\_b | Valor B en calendario | int |
| descripcion | Descripción de la tarea | int |
| Id\_estudiante | Llave foránea | int |

### 2.2.3. Interfaz de Usuario

#### **2.2.3.1. Calendario de Actividades**

La aplicación Follow Class contará con una vista para poder visualizar el calendario de actividades del usuario.

Interfaz de usuario gráfica

Descripción generada automáticamente

#### **2.2.3.2. Menú Agregar**

La aplicación Follow Class contará con una vista donde los usuarios podrán escoger el tipo de evento (clase o actividad) a agregar al calendario.

Interfaz de usuario gráfica

Descripción generada automáticamente

#### **2.2.3.3. Menú Agregar Curso**

La aplicación Follow Class contará con una vista donde los usuarios podrán escoger el tipo de evento (clase o actividad) a agregar al calendario.

Interfaz de usuario gráfica

Descripción generada automáticamente

#### **2.2.3.4. Menú Agregar Actividad**

La aplicación Follow Class contará con una vista donde los usuarios podrán agregar una actividad indicando el título, fecha. duración, prioridad, color, dar una etiqueta, y dar una descripción opcional.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### **2.2.3.5. Menú Editar Actividad**

La aplicación Follow Class contará con una vista donde los usuarios podrán editar una actividad y modificar los campos de título y etiqueta, logrando modificar también el color y la etiqueta respectivamente además de agregar una descripción.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente